# Advanced Methods for Model-based Diagnosis and Recovery

Complex spacecraft consist of several subsystems that exhibit different kinds of behavior including discrete and continuous. Uncertainty in several forms (sensor noise, uncertain environments etc.) make it difficult to determine the state of the system. We are developing a model-based diagnostic engine called Livingstone 3 that is capable of isolating discrete faults and parameter degradations and then synthesizing recovery actions that will allow the system to achieve mission goals.

Background

The Livingstone model-based diagnosis & recovery engine was used successfully in the remote agent experiment on Deep Space 1. Its successor, Livingstone 2 (L2), has been applied and is currently being applied to several test beds including the X-34 propulsion system (PITEX) and the International Space Station. In mid 2004 Livingstone 2 will participate in a flight experiment on Earth Observing Satellite 1 (EO-1). In the Livingstone on EO-1 (LEO-1) infusion experiment, L2 will be uploaded to EO-1 and will demonstrate monitoring and diagnosis of the operation of the spacecraft under command of the autonomy experiment.

Livingstone and L2 use a propositional logic representation of the model where variables are from a Boolean domain and behavior is expressed as clauses over these variables. It is possible to represent finite domain variables and equality or inequality constraints over these variables in the above framework. Livingstone then uses unit propagation to assign values to unknown variables using information about inputs/commands, observations and the current state of the system. If this results in an inconsistency, a conflict in the form of assumptions on component modes is generated and the generated conflicts are used to guide the search for new candidates.

The key limitation of L2 is that it allows only those models that can be converted to propositional logic. As a result, significant modeling effort has to be spend in abstracting the system behavior to fit in this framework. Additionally sophisticated monitors need to built to convert sensor data that may be in quantitative form to discrete form. Finally sensor noise and uncertainties are difficult to handle in this architecture.
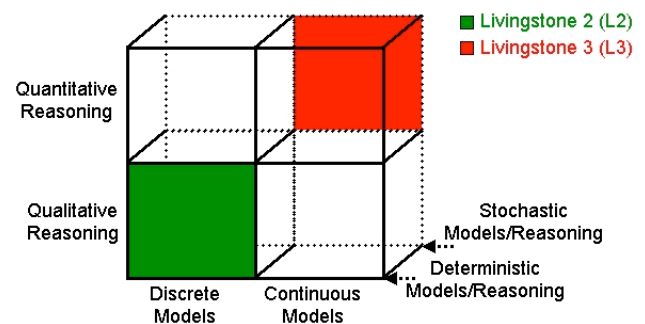
Research Overview

We are extending Livingstone's capabilities along three dimensions as illustrated in the cube figure. This diagnostic engine called Livingstone 3 uses a much more expressive modeling paradigm. Specifically, system models are comprised of three kinds of models: (i) the Component Connection model captures the various subsystems and components/devices in the system and the interaction between these in the form of connections, (ii) the Transition model captures the modes of operation of the components (including fault modes) and conditions for transitions between these modes, and (iii) the Behavioral model captures the behavior of components in each mode of operation.

The behavior model is expressed as constraints over variables from several different domains including Boolean, finite domain, real/interval valued domain, graphs and can take on the form of propositional formula, equality/inequality, differential algebraic equations and labeled edges.

Advanced Methods for Model-based Diagnosis & Recovery

The diagnostic engine consists of a candidate manager that keeps track of a set of most likely candidates given the commands issued to the system and the observations sensed from the system. For each candidate there is an hybrid observer that uses the transition model to track the modes of the components of the system. Constraint stores and constraint programming is used to track the evolution of behavior of the system for each candidate. When this tracking is inconsistent with the observations, justifications associated with the constraint programming are used to identify conflicts. A conflict-directed search procedure based on desired heuristics is used to generate new candidates which then continue to be tracked. One key characteristic is that both conflicts and candidates span time steps.

Recovery from a selected candidate can be effected by using an architecture similar to diagnostic engine. The idea is to still generate candidates. However, this time consistency is being checked against the desired goal observations. In addition candidates involve commands to fire commanded transitions as opposed to fault events that triggered autonomous fault transitions. The recovery engine comes up with a sequence of commands that is expected to push the system to the desired goal state.

In order to deal with uncertainties, the modeling paradigm is augmented by associating probabilities with transitions in the transition model. The behavior tracking for each candidate can then be done using a Bayesian approach (for example particle filters) in order to assign posterior probabilities to candidates as opposed to priors.

Parametric degradations can be diagnosed as a two step process where in the first step the deviating parameter is isolated using the Livingstone 3 approach and then in the second step parameter estimation techniques are used to estimate the extent of degradation.
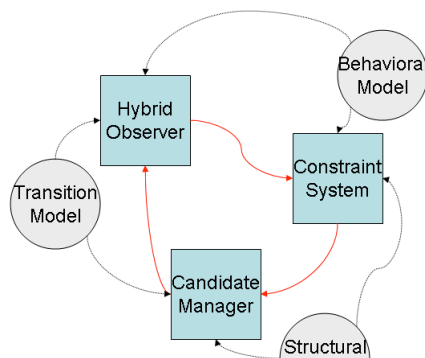
Relevance to Exploration Systems

Integrated system health management will be a key functionality of any exploration mission to ensure safety. We believe that model-based diagnosis & recovery techniques that cover the multitude of subsystems making up the exploration mission would facilitate fast efficient ISHM. The use and re-use of component models allow us to deal with long durations of missions, staged construction as well allow quick turn around of crew exploration vehicles. The ability to synthesize recovery actions facilitate on-board decision making autonomously or by humans. This will have a system of systems impact since

H&RT Program Elements:

This research capability supports the following H&RT program /elements:

ASTP/Software, Intelligent Systems & Modeling

Points of Contact:

Sriram Narasimhan
650 604 0832; sriram@email.arc.nasa.gov

Lee Brownston
650 604 2010; lbrownston@email.arc.nasa.gov